Project Design Phase-II

Technology Stack (Architecture & Stack)

|  |  |
| --- | --- |
| Date | 29 JUNE 2025 |
| Team ID | LTVIP2025TMID54832 |
| Project Name | BOOKNEST: Where stories nestle |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

**BookNest’s technical architecture** follows a multi-tier structure to ensure efficiency, scalability, and security. The **frontend** is built using technologies like HTML, CSS, and JavaScript (React.js), delivering a responsive user interface. The **backend** is powered by Node.js or Django, managing business logic, authentication, and API handling. **RESTful APIs** enable smooth communication between frontend and backend components. A **relational or NoSQL database** like MySQL or MongoDB stores books, users, and order data securely. **Payment integration** is handled using trusted gateways like Razorpay or Stripe. Cloud platforms such as **AWS or Firebase** support hosting and deployment. The system is designed to allow future scaling and feature upgrades with minimal disruption.

**Table-1: Components & Technologies**

|  |  |  |
| --- | --- | --- |
| **Test Component** | **Technology / Tool** | **Purpose** |
| UI Testing | Selenium, Cypress | To test user interface elements and workflows |
| Functional Testing | Postman, JUnit, Mocha | To validate all features like login, search, checkout |
| API Testing | Postman, REST Assured | To test RESTful API requests and responses |
| Performance Testing | JMeter, Lighthouse | To test page speed, load time, and stress handling |
| Security Testing | OWASP ZAP, Burp Suite | To check vulnerabilities like XSS, SQL injection |
| Cross-Browser Testing | BrowserStack, LambdaTest | To test website compatibility on different browsers |
| Mobile Responsiveness | Chrome DevTools, Responsinator | To ensure UI adapts to various screen sizes |
| Test Management | JIRA, TestRail | For managing test cases, reporting bugs, and tracking |
| Automation Testing | Selenium WebDriver, Cypress | To automate repetitive test scenarios |
| CI/CD Integration | GitHub Actions, Jenkins | To run tests automatically on code push/deploy |

**Table-2: Application Characteristics**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Characteristic** | **Description** |
| 1 | User-Friendly Interface | Simple and intuitive UI for easy navigation and smooth user experience |
| 2 | Responsiveness | Works seamlessly across desktops, tablets, and mobile devices |
| 3 | Scalability | Supports increasing numbers of users, books, and transactions efficiently |
| 4 | Security | Secures user data, transactions, and login sessions using best practices |
| 5 | Personalization | Offers recommendations and tailored content based on user preferences |
| 6 | Performance Optimization | Fast load times and efficient search and filter operations |
| 7 | Reliability | Ensures minimal downtime and stable functioning of all core features |
| 8 | Maintainability | Modular codebase makes it easy to update and extend features |
| 9 | Integration Support | Easily integrates with third-party APIs like payment gateways and email tools |